SimNet: Enabling Robust Unknown Object Manipulation from Pure Synthetic Data via Stereo

Anonymous Author(s)
Affiliation
Address
email

Abstract: Robot manipulation of unknown objects in unstructured environments is a challenging problem due to the variety of shapes, materials, arrangements and lighting conditions. Even with large-scale real-world data collection, robust perception and manipulation of transparent and reflective objects across various lighting conditions remains challenging. To address these challenges we propose an approach to performing sim-to-real transfer of robotic perception. The underlying model, SimNet, is trained as a single multi-headed neural network using simulated stereo data as input and simulated object segmentation masks, 3D oriented bounding boxes (OBBs), object keypoints and disparity as output. A key component of SimNet is the incorporation of a learned stereo sub-network that predicts disparity. SimNet is evaluated on 2D car detection, unknown object detection and deformable object keypoint detection and significantly outperforms a baseline that uses a structured light RGB-D sensor. By inferring grasp positions using the OBB and keypoint predictions, SimNet can be used to perform end-to-end manipulation of unknown objects in both "easy" and "hard" scenarios using our fleet of Toyota HSR robots in four home environments. In unknown object grasping experiments, the predictions from the baseline RGB-D network and SimNet enable successful grasps of most of the "easy" objects. However, the RGB-D baseline only grasps 35% of the "hard" (e.g., transparent) objects, while SimNet grasps 95%, suggesting that SimNet can enable robust manipulation of unknown objects, including transparent objects, in unknown environments. Additional visualizations and materials are located at https://tinyurl.com/simnet-corl.

Keywords: Sim-to-Real, Computer Vision, Manipulation

1 Introduction

To successfully deploy robots into diverse, unstructured environments such as the home, robots must have robust and general behaviors that that can adapt to variations in lighting, furniture, and objects. As such, robots must be able to to perceive and manipulate objects that they have not seen before. This is made more challenging by the fact that many everyday objects are made of reflective or transparent materials and are located in places that have harsh lighting (e.g., Figure 3, right side).

To perform well in these settings, large-scale real-world data collection is often performed [1, 2, 3], which is costly, time-consuming, has limited flexibility and sometimes contains missing data (as is the case with RGB-D sensing of transparent objects). An alternative is to use simulation to automatically label a large dataset of images [2, 4, 5, 6, 7, 8, 9, 10, 11, 12]. Although physics-based and photorealistic simulations can enable end-to-end learning methods, rendering can take significant computational resources and artists must generate high-quality artefacts [2, 13].

This paper proposes an approach for sim-to-real transfer of robot perception and manipulation that does not require large-scale real-world datasets to be collected and does not require photorealism during simulation. The approach has three primary components. First, a non-photorealistic simulator that randomizes over lighting and textures is used to automatically produce large-scale domain-randomized data that includes stereo images [7, 14], 3D oriented bounding boxes, object keypoints, and segmentation masks (Figure 1). The second component and a core contribution SimNet (Figure 2) is a lightweight multi-headed neural network that is trained exclusively on simulated data and takes as input stereo pairs and predicts segmentation masks, 3D oriented bounding boxes (OBBs), keypoints

Submitted to the 5th Conference on Robot Learning (CoRL 2021). Do not distribute.

Figure 1: **Simulation images:** Simulated data was generated for three domains: cars, small objects and t-shirts using a non-photorealistic simulator with domain-randomization. Dataset generation is parallelized across machines and can be generated in an hour for $60 (USD) cloud compute cost. Labels for OBBs, semantic segmentation masks, depth images, and keypoints can automatically be generated. The data is used to train SimNet. By learning to focus on geometry, sim2real transfer can be performed using only very low-quality scenes.

and disparity as output. Key to our approach is the ability to predict disparity images using a differentiable cost volume to match features in a stereo image [15, 16]. This network is end-to-end trained using a depth reconstruction auxiliary loss, which encourages the network to focus on disparity predictions and geometric information that are task-relevant. Unlike other approaches, which use active depth sensing [2, 4, 5, 6, 8], the incorporation of the learned differentiable stereo network into SimNet enables perception in the presence of non-Lambertian surfaces such as reflective or transparent materials and in challenging lighting conditions such as direct sunlight. SimNet can run at 20Hz on a Titan Xp GPU. The third component is a heuristic method for predicting grasp positions from the OBB and keypoint predictions. These grasp positions are combined with a classical planner to execute manipulation behaviors [5, 10, 17, 18, 19].

SimNet is evaluated on three real-world domains: 2D car detection, unknown object grasping and deformable object keypoint prediction for t-shirt folding. SimNet is trained on a large-scale synthetic dataset generated for each domain and evaluated on real-world validation sets that include optically-challenging objects, distractor objects, furniture and direct sunlight. The baseline network predicts the same outputs, but uses RGB-D input from a structured light sensor. All three domains showed mAP gains with SimNet over the baseline. SimNet is also evaluated in an unknown object grasping experiment with both "easy" and "hard" objects using a fleet of Toyota HSR-C robots in four different homes. The "hard" objects are optically challenging and are reflective or transparent. Both the RGB-D network and SimNet are able to effectively generate grasps for the easy objects. However, the robot is only able grasp 35% of the "hard" objects with the RGB-D network, while it is able to grasp 95% with SimNet. These results demonstrate that SimNet is able to grasp a wide variety of unknown objects, including transparent objects, in unstructured environments. SimNet is also evaluated on a t-shirt folding experiment across several homes and shirts.

This work makes five contributions: ***(i)*** an efficient neural network, SimNet, that leverages learned stereo matching to enable robust multi-headed predictions of keypoints, oriented bounding boxes (OBBs), semantic segmentation masks, and depth images using only low-quality synthetic data, ***(ii)*** the first network to enable single-shot prediction of 3D oriented bounding boxes of unknown objects ***(iii)*** stereo vision experiments on simulated and real home scenes with annotations for oriented bounding boxes, segmentation masks, depth images, and keypoints, ***(iv)*** experiments on the KITTI stereo vision benchmark dataset that suggest that the network transfers significantly better than standard domain randomization and naive stereo concatenation, and ***(v)*** experiments in four homes that suggest that the network robustly transfers to real-world environments compared to active depth sensing and can be used to construct robot manipulation policies for grasping and t-shirt folding. The model training code and validation datasets will be publicly released.

2 Related Work

Our related work is divided into perception for manipulation, sim-to-real and learned stereo.

2.1 Parameterized Representations for Manipulation

In contrast to end-to-end robot policy learning [3, 20, 21, 22], a popular approach for generating behaviors is to parameterize motions by the outputs of a perception network [4, 5, 10, 17, 23]. This decouples perception from planning and control, and enables perception systems to be trained in simulation without the need for accurate physical simulation. One representation commonly used for

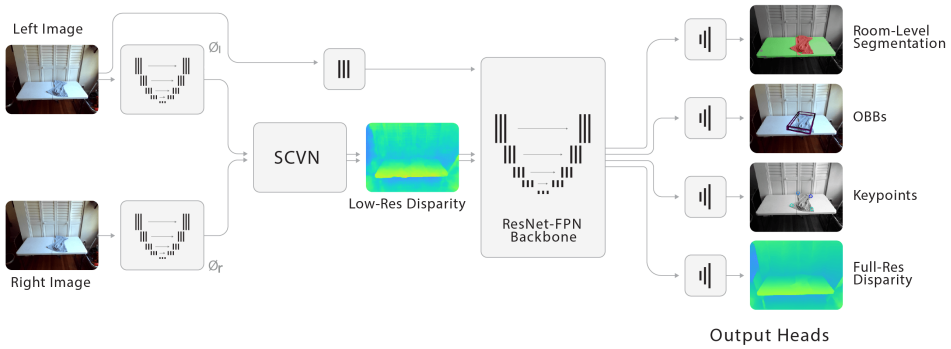

Figure 2: **SimNet:** In SimNet, each stereo RGB image is fed into a feature extractor before being fed into SCVN, which performs approximate stereo matching. The output of the SCVN is a low resolution disparity image fed in with features from the left image to a ResNet-FPN backbone and output prediction heads. The output heads predict room-level segmentation, OBBs, keypoints, and full-resolution disparity images.

grasping of rigid objects are oriented bounding boxes (OBBs) [23]. These are useful for grasping common household objects, by aligning grasps along the principal components [23]. Our work contributes a method to directly predict OBBs from a neural network. A representation that is used for deformable manipulation are keypoints or learned correspondences, which have seen significant success in tasks such as deformable manipulation [5, 10, 12, 17, 18, 24] and grasping [25, 26]. Ganapathi et al. [10] and Sundaresan et al. [5] predict correspondences from domain-randomized depth or monocular RGB images and demonstrate impressive transfer to real fabrics and ropes in constrained lab environments. In contrast, we demonstrate robust transfer on diverse home scenes, and compare to the methods in these papers as baselines.

2.2 Sim-to-Real Transfer

Transferring perception models trained on simulated data to reality is an active area of research. One technique to transfer from simulation to reality is to train directly on a geometric representation of the world such as a point cloud or depth image [4, 5]. Prior work has used depth-based transfer techniques in a variety of scenarios, such as grasping unknown objects [4], tying a rope [5], and segmenting object instances in cluttered scenes [6, 8]. Depth-based transfer is commonly performed with an active sensor using structured light. The applicability of these sensors is limited to regimes where their reflected infrared pattern is consistent enough that it can be pattern matched for 3-D reconstruction [27]. However, home settings contain non-Lambertian objects such as glassware and a large amount of natural light, which can create large sensing errors. Existing work studies combining depth sensors with RGB information to fill in missing depth information for clear objects [2] using photorealistic simulation and real data. In contrast to [2], we exclusively use inexpensive RGB sensors and low quality simulated data for training. Our approach is comparable to Xie et al. [6], which separately leverages depth and synthetic RGB inputs to compensate for the limitations of depth sensing alone. We compare to several variants of Xie et al. [6] as a baseline approach, and find that using stereo images outperforms using RGB and depth on optically challenging objects.

An alternative to depth-based transfer is domain randomization with RGB images (DR) [7, 14, 28, 29]. Typically, DR randomizes over the lighting and textures in the environment by utilizing a low-quality, non-photorealistic renderer and has produced successful results in car detection [28], 6-DOF pose estimation of known objects [30], and camera calibration [31]. By injecting large randomization into texture and lighting, the network is forced to use the geometry of the scene to solve the task [14]. A limitation of this approach is that geometric reasoning on a monocular image requires leveraging global shape priors, lighting effects, and scene understanding [32]. In this work, we propose a transfer technique that uses learned stereo matching with domain randomization to extract geometric features using only local non-semantic information and demonstrate robustness on the KITTI benchmark.

2.3 Learned Stereo Matching

The goal of stereo matching is to try to explicitly compute the pixel displacement offset between objects from the left and right images in a stereo pair [33]. By computing a per-pixel displacement offset, or disparity, a high-resolution point cloud can be generated of the world. Given that it is challenging to estimate similarity features of image patches, recent work has explored using deep

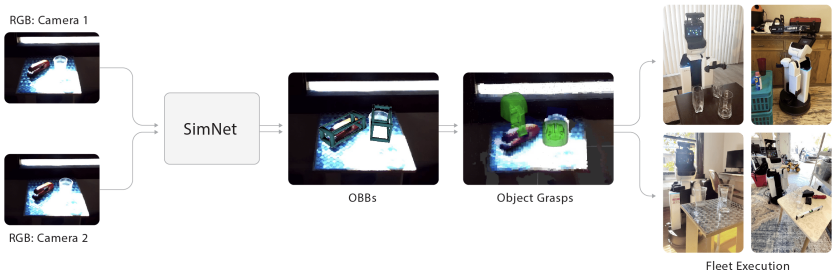

Figure 3: **Manipulation overview**: A stereo pair is fed as input to SimNet, which produces OBBs. Grasp positions are produced from OBBs and are used by a classical planner to grasp the object [23]. This grasping technique has been deployed on a fleet of home robots to perform manipulation in optically challenging scenarios.

learning techniques to advance stereo techniques [15, 16, 34, 35, 36, 37]. In contrast to the learned stereo community that targets achieving perfect depth-sensing when training with real data [15, 16], we are interested in enabling transfer from simulation for "high-level" vision tasks. We specifically propose a network for manipulation, SimNet, that regresses coarse depth with an efficient stereo matching network architecture based on [15, 16] to learn features that are robust enough to predict oriented bounding boxes, segmentation masks, and keypoints on real objects for manipulation.

3 SimNet: Enabling Predictions for Manipulation From Synthetic Stereo

In this section, we will discuss our network architecture that leverages approximate stereo matching techniques from [15, 16] and domain randomization [7, 14] to predict segmentation masks, OBBs, and keypoints on unseen objects for robot manipulation. Our key insight is that by training approximate stereo matching algorithms on pure synthetic data, we can learn robust "low-level" features like disparity than enable sim2real transfer of "high-level" vision tasks. We first discuss how we learn robust low-level features and then feed that into high-level perception algorithms. Finally, we discuss how our low-cost synthetic data is generated. For training details see supplemental Sec. B.3; the overall network architecture is in Fig. 2.

3.1 Stereo Cost Volume Networks (SCVN) For Robust Low-Level Features

We will now describe the stereo matching neural network module used in SimNet's backbone, which enables transfer from simulation. In this section, we use $\odot$ to denote Hadamard products, and use $I_{[i,j:k,:]}$ to select all elements with index i in the first dimension of tensor I, index in $\{j, \ldots k-1\}$ in the second dimension of I, and any index in the third dimension onwards. Let I_{l} and I_{r} denote the left and right RGB images from the stereo pair. Each image has dimension $3 \times H_0 \times W_0$. The left and right images are fed into neural networks Φ_{l} and Φ_{r} that featurize each image respectively and output feature volumes ϕ_{l} and ϕ_{r}. Both ϕ_{l} and ϕ_{r} have dimension $C_\phi \times H_\phi \times W_\phi$, where C_ϕ is the number of channels in each feature volume, and H_ϕ and W_ϕ are their height and width, respectively. We used a lightweight Dilated ResNet-FPN [38] as our feature extractor, to enable large receptive fields with a minimal amount of convolutional layers.

The extracted features ϕ_{l} and ϕ_{r} are fed into a stereo cost volume network f_{cost} that consists of an approximate stereo matching module that searches horizontally in the feature volumes for correspondences within an allowed disparity range. In classical stereo vision literature, correspondences across left and right images can be found by searching along a horizontal line across the images for a match, and the disparity is the difference in the x coordinates in the match, which is high for closer points in 3D space and low for farther points. The architecture for f_{cost} is heavily inspired by the techniques in [15] and approximately performs this search. The first phase of the network $f^{(0)}_{\mathrm{cost}}$ computes pixelwise dot products between horizontally shifted versions of the feature volumes. The output of this phase has dimension $C_{\mathrm{c}} \times H_\phi \times W_\phi$. $2 * (C_{\mathrm{c}} - 1)$ represents the maximum disparity considered by the network, and the minimum disparity considered is 0. The i-th $H_{\mathrm{c}} \times W_{\mathrm{c}}$ slice of the output is computed as:

$$f^{(0)}_{\mathrm{cost}}(\phi_{\mathrm{l}}, \phi_{\mathrm{r}})_{[i,:,:i:]} = \sum_{j=0}^{C-1} \left(\phi_{\mathrm{l},[\mathrm{i},:,\mathrm{i}:]} \odot \phi_{\mathrm{r},[\mathrm{i},:,:\mathrm{W}-\mathrm{i}]}\right)_{[j]}; \qquad f^{(0)}_{\mathrm{cost}}(\phi_{\mathrm{l}}, \phi_{\mathrm{r}})_{[i,:,:i]} = 0$$

The first case takes the rightmost $H_c - i$ columns of the left feature volume ϕ_l and computes a pixelwise dot product with the leftmost $H_c - i$ columns of ϕ_r. This operation horizontally searches for matches across the two feature volumes at a disparity of $2i$. The next phase of the network $f_{\text{cost}}^{(1)}$ feeds the resulting volume into a sequence of ResNet blocks, which outputs a volume of dimension $C_c \times H_\phi \times W_\phi$ before performing a soft argmin along the first axis of the volume. The soft argmin operation approximately finds the disparity for each pixel by locating its best match. The final volume is an estimate of a low-resolution disparity image $\hat{I}_{\text{d,low}}$ with shape $H_\phi \times W_\phi$. We denote $f_{\text{cost}} = f_{\text{cost}}^{(1)} \circ f_{\text{cost}}^{(0)}$.

Disparity Auxiliary Loss In addition to the losses for the high-level perception heads, we additionally train the weights of Φ_l, Φ_r, and f_{cost} by minimizing an auxiliary depth reconstruction loss function. In particular, the loss function takes in a target disparity image $I_{\text{targ,d}}$ of dimension $H_0 \times W_0$, downsamples it by a factor of H_0/H_ϕ and then computes the Huber loss [39] $\ell_{\text{d,small}}$ of it with the low-resolution depth prediction $f_{\text{cost}}(\phi_l, \phi_r)$. That is, the network weights are trained to minimize $\ell_{\text{d,small}}(f_{\text{cost}}(\phi_l, \phi_r), \texttt{downsample}(I_{\text{targ,d}}, H_0/H_\phi))$.

3.2 Extracting High-Level Predictions for Manipulation Tasks

Given a SCVN to extract geometric features from stereo images, we need to learn on top of it high-level predictions relevant to manipulation. To design a backbone for robust simulation-trained manipulation, we feed the output of a SVCN, $\hat{I}_{\text{d,low}}$, into a Resnet18-FPN [40] feature backbone f_{backbone}. Additionally, we leverage early stage features from the left RGB image, I_l, to enable high resolution texture information to be considered at inference time, similar to [6]. The features are extracted from the ResNet stem and concatenate with the output of the SCVN and fed into the backbone. The output of the backbone is fed into each of the prediction heads. In this section, we describe each of the prediction heads of SimNet (Fig. 2) and the losses used to train it. Each prediction head high uses the up-scaling branch defined in [40], which aggregates different resolutions across the feature extractor.

Room Level Segmentation: Given a mobile robot that is trying to manipulate objects on a table, it is useful to be able to know where the table is in the room and the objects on the table. We can express this level of scene understanding as segmentation problem consisting of predicting three categories: surfaces, objects and background. Cross-entropy loss l_{seg} is used for training, as in prior work [40].

Oriented Bounding Boxes: Detection of an OBB requires determining individual object instances as well as estimating translation, $t \in \mathbb{R}^3$, scale $S \in \mathbb{R}^{3\times3}$, and rotation, $R \in \mathbb{R}^{3\times3}$, of the encompassing OBB. We can recover these parameters by using four different output heads. First to recover object instances, we regress a $W_0 \times H_0$ image,which is the resolution of the input left image, where for each object in the image a Gaussian heatmap is predicted. Instances can then be derived using peak detection as in [41, 42]. We use an L_1 loss on this output head and denote the loss as l_{inst}.

Given instances of object, we can now regress the remaining 9-DOF pose parameters. To recover scale and translation, we first regress a $W_0/8 \times H_0/8 \times 16$ output head where each element contains pixel-wise offset from detected peak to the 8 box vertices projected on to the image. We can then recover the scale and translation of the box up to a scale ambiguity using EPnP similar to [41]. In contrast to prior work in pose estimation, we aligned the predicted box based on principal axes sized in a fixed reference frame. To recover absolute scale and translation, we additionally regress the distance from the camera $z \in \mathbb{R}$ of the box centroid as a $W_0/8 \times H_0/8$ tensor. The two losses on these tensors are an L_1 loss and are denoted l_{vrtx} and l_{cent}.

Finally the rotation, R of the OBB can be recovered via directly predicting the covariance matrix, $\Sigma \in \mathbb{R}^{3\times3}$ of the ground truth 3D point cloud of the target object, which can be easily generated in simulation. We directly regressed an output tensor of $W_0/8 \times H_0/8 \times 6$, which for each pixel contains both the diagonal and symmetric off diagonal elements of the target covariance matrix. We can then recover rotation based on the **SVD** of Σ. We additionally use an L_1 loss on this output head and denote the loss l_{rot}. Note for the 9-DOF pose loses, we only enforce them where the Gaussian heatmaps are greater than 0.3, to prevent ambiguity in empty space.

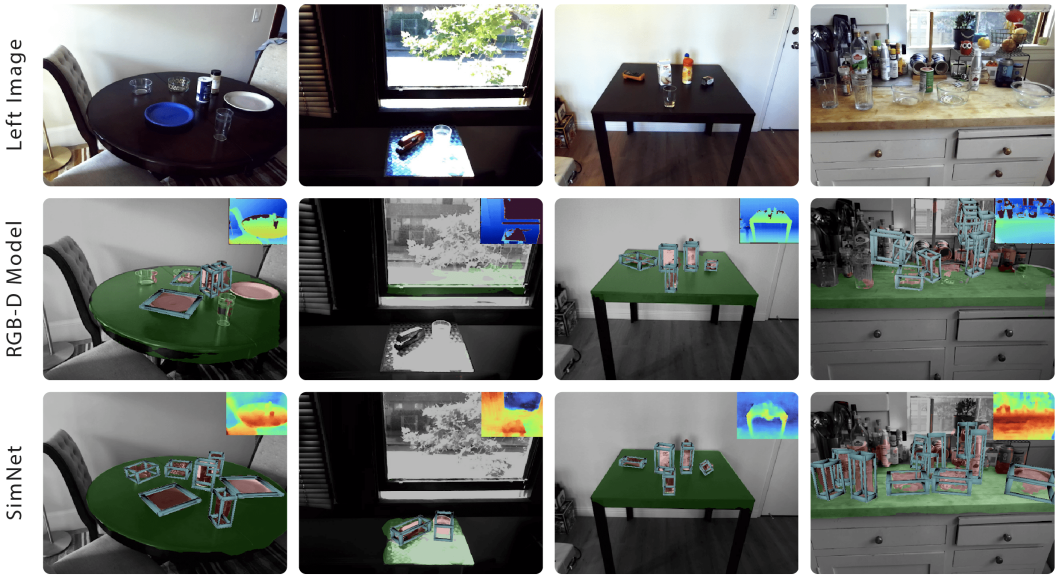

Figure 4: **Graspable Object Predictions:** SimNet is evaluated on an oriented bounding box regression task with objects of varying sizes and shapes placed on flat surfaces. Top: Is the left RGB image from different homes. Middle: Is our model trained with RGB-D sim2real transfer similar to [6]. The top right corner is the output of the Asus Xtion depth sensor. Bottom: Is SimNet. In the top right, we show the low-res disparity estimate predicted by the learned stereo net. SimNet consistently enables better sim2real transfer of the predictions for optically challenging scenarios.

Keypoints: Keypoints and learned correspondences are a common representation for robot manipulation, especially in deformable manipulation [5, 10, 17, 18, 25]. SimNet has an output head that predicts keypoints of various classes, which can be fed into planners for manipulation. For instance, one of the keypoint classes we predict in Sec. 4.1 are t-shirt sleeves, which is used for t-shirt folding. The output head predicts heatmaps for each keypoint class, and is trained to match target heatmaps with Gaussian distributions placed at each ground-truth keypoint location using a pixelwise cross-entropy loss l_{kp}. To extract keypoints from the predicted heatmaps, we use non-maximum suppression to perform peak detection as in Hou et al. [41].

Full Resolution Disparity: The final output head is a full resolution disparity image, which can be converted into a 3D point cloud for collision avoidance. Since our SCVN, produces a disparity image at quarter resolution, we can combine the feature extractor backbone and the left stereo image to produce a full resolution depth image. We use the same branch architecture as the previous heads to aggregate information across different scales. During training we use the same loss as the SCVN, but enforced at full resolution. This output is trained using a Huber loss function.

3.3 Efficient Synthetic Dataset Generation

Given the complexity of all the output predictions defined in the previous section, it would be impractical to label a sufficient amount of real-data to generalize across scenes. Thus, we are interested in using synthetic data to provide ground truth annotations on a wide variety of scenarios. To force the network to learn geometric features, we randomize over lighting and textures as recommended in Mayer et al. [36]. In contrast to [2, 13], we use OpenGL shaders instead of physically based rendering [43] approaches. Low-quality rendering greatly speeds up computation, and allows for data-set generation on the order of an hour. We generate three datasets: cars, graspable objects, and t-shirts (Fig. 1), and discuss specific dataset details in supplemental Sec. D. We intend to release our datasets as well as real-world validation data as part of this paper.

4 Experiments

SimNet is evaluated on a set of real-world computer vision and robotics tasks to see if training on synthetic stereo data will transfer robustly to diverse, real images in unstructured environments and enable manipulation of optically challenging objects. SimNet is compared against baseline

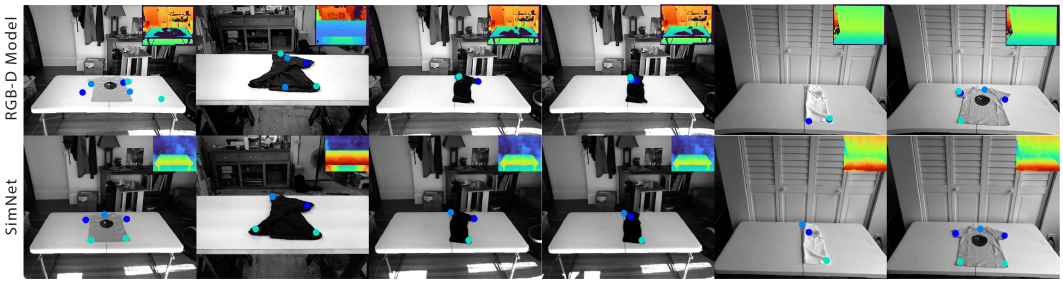

Figure 5: **T-shirt Keypoint Predictions:** SimNet is evaluated on keypoint regression for shirts in various stages of folding. Three classes of keypoints are predicted: sleeves, neck, and bottom corners. RGB-D performs poorly due to strong natural lighting and minimal depth variation. SimNet accurately predicts keypoints on the shirts despite these challenges.

approaches using monocular, naive stereo concatenation and active RGB-D sensing (supplemental Sec. B.2).

4.1 Manipulation Experiments

The learned perception models and manipulation policies are evaluated on two optically challenging tasks: (1) grasping objects on tabletops and (2) t-shirt folding.

4.1.1 Experimental Setup

All physical experiments are conducted on tabletops found across four different, real homes. Manipulation is performed with the Toyota HSR robot, which has a four DOF arm, mobile base, and pair of parallel jaw grippers [44]. It has a mounted stereo pair from a Zed 2 camera and a Asus Xtion Pro RGB-D sensor. Each home has different background objects, furniture, graspable objects, and lighting conditions, which evaluates each network's ability to robustly generalize to diverse scenarios.

4.1.2 Ablation of Baselines on Validation Data

We compare to the following baselines: **mono**, which uses the left stereo image, **depth**, which only uses depth inputs, and **RGB-D**, which uses both RGB and depth inputs. To make the RGB-D baseline competitive, we swept over four different algorithms to fuse color and depth information on a held out set of 500 non-optically challenging scenes (i.e. matte objects with minimal natural light). We measured 3D mAP@0.25, a common metric in pose prediction [45], of annotated bounding boxes of each object on the tabletop of interest. We report the best results of each method in Table 1. Additional sensing ablations can be found in Sec. C.

Method	Mono	Depth	RGB-D	SimNet
3D mAP	0.164	0.831	0.855	**0.921**

Table 1: **SimNet Perception Results:** We present an ablation of different sensing modalities on the 3D OBB prediction task. We evaluate 3D mAP on a dataset of real, human-annotated images of optically **easy objects.** We find that monocular sensing performs poorly on this task, likely due to the task being 3D in nature, while depth and RGB-D perform much better. SimNet outperforms these methods on the real images. Baseline implementation details can be found in Sec. B.2. We present full ablation results in supplemental Sec. C.

4.2 Unknown Object Grasping

In this experiment, the robot's task is to grasp objects on a tabletop comprised of two classes of household objects in each home: **optically easy**, which consists of opaque, non-reflective objects, and **optically hard**, which contains optically challenging, transparent objects (Fig. 6). For each trial, we select an object uniformly at random from the dataset and randomly place it on the tabletop with other distractor objects. The task is to grasp the foremost object in the scene, by using a heuristic grasp planner that takes OBBs as input. Specifically, given an OBB, the robot aligns the gripper with the largest principal axis. In the event, of similar sized principal axes like a ball, the robot favors grasping the object on the side closest to the robot. A grasp is successful if the robot is able to raise the object off the table and remove it from the scene.

For each of the four homes we test five easy objects and five hard objects and compare SimNet against the best RGB-D baseline found in Sec. 4.1.2. We report quantitative results in Table 2 and qualitative results of the predictions in Fig. 4. SimNet outperforms RGB-D, 92.5% vs. 62.5% in grasp success.

Method (Object Class)	Home 1	Home 2	Home 3	Home 4	Overall
RGB-D (O. Easy)	4/5	**5/5**	4/5	**5/5**	**18/20**
SimNet (O. Easy)	**5/5**	4/5	**5/5**	4/5	**18/20**
RGB-D (O. Hard)	0/5	1/5	1/5	**5/5**	7/20
SimNet (O. Hard)	**5/5**	**5/5**	**5/5**	4/5	**19/20**

Table 2: **SimNet Grasping Results:** Grasp success scores between the best RGB-D method and SimNet across homes. On optically easy objects (i.e. matte and non-reflective) RGBD-D and SimNet perform similar on average. However, when presented with challenging objects such as glassware, SimNet outperforms RGB-D.

4.3 Grasp Point Prediction in T-shirt Folding

The robot is also evaluated on a t-shirt folding task, where it must execute a sequence of four folds on unseen, real t-shirts. This task is challenging to perform using depth sensing, because the depth resolution of most commercial depth sensors cannot capture the subtle variations in depth due to the thickness of a t-shirt. Keypoints are a popular representation for manipulating deformable objects [17, 18]. We parameterize a robot shirt folding policy using keypoint predictions for the shirt's neck, sleeves, and bottom corners (Sec. B.7.2). To compute quantitative results on sim2real transfer, we collect a validation dataset of 32 real images from 12 t-shirts in 3 homes of stages of t-shirt folding (Fig. 5) and report keypoint prediction mAP (Table 3). We additionally present videos of folding using the robot on the project website for qualitative evaluation. We find that SimNet significantly outperforms RGB-D and depth, and slightly outperforms mono. A large part of our dataset was collected in a room with direct natural light, which increase noise in the active depth sensor, due to large amount of infrared light.

Method	Mono	Depth	RGB-D	SimNet
mAP	0.893	0.282	0.631	**0.917**

Table 3: **T-Shirt Folding Results:** A comparison of keypoint mAP shows that active depth-based models transfer poorly on this task, due to interference from natural lighting and low depth profile of the shirts. The monocular network transfers well to the real shirt images, and SimNet slightly outperforms it. Unlike the models that use active depth sensing, SimNet is robust to lighting variation and the low depth variation of the shirts.

4.4 KITTI 2D Car Detection

In this experiment, we evaluate whether SimNet is effective sim-to-real for other vision tasks outside of home robotics such as 2D car detection on the KITTI benchmark [46, 47]. We compare the stereo cost volume network trained on simulated images to networks using monocular inputs (**Mono**), stacking left and right image inputs (**Stacked**), and using real data with a standard validation/training split used in [42] (**SimNet-real**). To predict 2D bounding boxes, we modify SimNet to use the single-shot box detection output head in CenterNet [42]. In the table below we report mAP@0.5 for the aggregate of moderate and easy classes car classes.

Interestingly, SimNet can significantly outperform monocular images and naive stereo concatenation, which suggests explicit stereo matching leads to more robust transfer. Furthermore, the gap between real and sim data is only 3.5%, which suggests relatively robust transfer. However, qualitatively SimNet fails to detect cars at distance, since the ability to reason about geometry decays with distance from the camera. For visualizations of the predictions see supplemental Fig. 13. Thus, SimNet is best used for applications with limited range from the camera, such as robot manipulation.

Method	Mono	Stacked	SimNet	SimNet-real
mAP	0.565	0.710	**0.826**	0.861

Table 4: **KITTI Results:** Results of different sim-to-real techniques for 2D car detection on the KITTI Benchmark. We find that the simply stacking the left and right images in the stacked input network significantly outperforms the monocular network. SimNet, however, transfers much better than naive stereo concatenation and is close to performance using real data.

5 Conclusion

We present SimNet, am efficient, multi-headed prediction network that leverages approximate stereo matching to transfer from simulation to reality. The network is trained on entirely simulated data and robustly transfers to real images of unknown optically-challenging objects such as glassware, even in direct sunlight. We show that these predictions are sufficient for robot manipulation such as t-shirt folding and grasping. In future work, we will plan to use SimNet to automate household chores.

References

[1] A. Gupta, A. Murali, D. Gandhi, and L. Pinto. Robot learning in homes: Improving generalization and reducing dataset bias. *arXiv preprint arXiv:1807.07049*, 2018.

[2] S. Sajjan, M. Moore, M. Pan, G. Nagaraja, J. Lee, A. Zeng, and S. Song. Clear grasp: 3d shape estimation of transparent objects for manipulation. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3634–3642. IEEE, 2020.

[3] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke, et al. Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation. *arXiv preprint arXiv:1806.10293*, 2018.

[4] J. Mahler, M. Matl, V. Satish, M. Danielczuk, B. DeRose, S. McKinley, and K. Goldberg. Learning ambidextrous robot grasping policies. *Science Robotics*, 4(26), 2019.

[5] P. Sundaresan, J. Grannen, B. Thananjeyan, A. Balakrishna, M. Laskey, K. Stone, J. E. Gonzalez, and K. Goldberg. Learning rope manipulation policies using dense object descriptors trained on synthetic depth data. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9411–9418. IEEE, 2020.

[6] C. Xie, Y. Xiang, A. Mousavian, and D. Fox. The best of both modes: Separately leveraging rgb and depth for unseen object instance segmentation. In *Conference on robot learning*, pages 1369–1378. PMLR, 2020.

[7] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 23–30. IEEE, 2017.

[8] M. Danielczuk, M. Matl, S. Gupta, A. Li, A. Lee, J. Mahler, and K. Goldberg. Segmenting unknown 3d objects from real depth images using mask r-cnn trained on synthetic data. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 7283–7290. IEEE, 2019.

[9] I. Akkaya, M. Andrychowicz, M. Chociej, M. Litwin, B. McGrew, A. Petron, A. Paino, M. Plappert, G. Powell, R. Ribas, et al. Solving rubik's cube with a robot hand. *arXiv preprint arXiv:1910.07113*, 2019.

[10] A. Ganapathi, P. Sundaresan, B. Thananjeyan, A. Balakrishna, D. Seita, J. Grannen, M. Hwang, R. Hoque, J. E. Gonzalez, N. Jamali, et al. Learning dense visual correspondences in simulation to smooth and fold real fabrics. *arXiv preprint arXiv:2003.12698*, 2020.

[11] R. Hoque, D. Seita, A. Balakrishna, A. Ganapathi, A. K. Tanwani, N. Jamali, K. Yamane, S. Iba, and K. Goldberg. Visuospatial foresight for multi-step, multi-task fabric manipulation. *arXiv preprint arXiv:2003.09044*, 2020.

[12] D. Seita, A. Ganapathi, R. Hoque, M. Hwang, E. Cen, A. K. Tanwani, A. Balakrishna, B. Thananjeyan, J. Ichnowski, N. Jamali, et al. Deep imitation learning of sequential fabric smoothing from an algorithmic supervisor. *arXiv preprint arXiv:1910.04854*, 2019.

[13] M. Roberts and N. Paczan. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. *arXiv preprint arXiv:2011.02523*, 2020.

[14] F. Sadeghi and S. Levine. Cad2rl: Real single-image flight without a single real image. *arXiv preprint arXiv:1611.04201*, 2016.

[15] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4040–4048, 2016.

[16] A. Kendall, H. Martirosyan, S. Dasgupta, P. Henry, R. Kennedy, A. Bachrach, and A. Bry. End-to-end learning of geometry and context for deep stereo regression. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 66–75, 2017.

[17] J. Grannen, P. Sundaresan, B. Thananjeyan, J. Ichnowski, A. Balakrishna, M. Hwang, V. Viswanath, M. Laskey, J. E. Gonzalez, and K. Goldberg. Untangling dense knots by learning task-relevant keypoints. *arXiv preprint arXiv:2011.04999*, 2020.

[18] D. Seita, N. Jamali, M. Laskey, A. K. Tanwani, R. Berenstein, P. Baskaran, S. Iba, J. Canny, and K. Goldberg. Deep transfer learning of pick points on fabric for robot bed-making. *arXiv preprint arXiv:1809.09810*, 2018.

[19] P. Sundaresan, B. Thananjeyan, J. Chiu, D. Fer, and K. Goldberg. Automated extraction of surgical needles from tissue phantoms. In *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*, pages 170–177. IEEE, 2019.

[20] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel, et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018.

[21] B. Thananjeyan, A. Balakrishna, S. Nair, M. Luo, K. Srinivasan, M. Hwang, J. E. Gonzalez, J. Ibarz, C. Finn, and K. Goldberg. Recovery rl: Safe reinforcement learning with learned recovery zones. *IEEE Robotics and Automation Letters*, 6(3):4915–4922, 2021.

[22] S. Levine, C. Finn, T. Darrell, and P. Abbeel. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1):1334–1373, 2016.

[23] R. Balasubramanian, L. Xu, P. D. Brook, J. R. Smith, and Y. Matsuoka. Physical human interactive guidance: Identifying grasping principles from human-planned grasps. *IEEE Transactions on Robotics*, 28 (4):899–910, 2012.

[24] J. Maitin-Shepard, M. Cusumano-Towner, J. Lei, and P. Abbeel. Cloth grasp point detection based on multiple-view geometric cues with application to robotic towel folding. In *2010 IEEE International Conference on Robotics and Automation*, pages 2308–2315. IEEE, 2010.

[25] L. Manuelli, W. Gao, P. Florence, and R. Tedrake. kpam: Keypoint affordances for category-level robotic manipulation. *arXiv preprint arXiv:1903.06684*, 2019.

[26] P. R. Florence, L. Manuelli, and R. Tedrake. Dense object nets: Learning dense visual object descriptors by and for robotic manipulation. *arXiv preprint arXiv:1806.08756*, 2018.

[27] A. Grunnet-Jepsen, J. N. Sweetser, P. Winer, A. Takagi, and J. Woodfill. Projectors for intel® realsense™ depth cameras d4xx. *Intel Support, Interl Corporation: Santa Clara, CA, USA*, 2018.

[28] J. Tremblay, A. Prakash, D. Acuna, M. Brophy, V. Jampani, C. Anil, T. To, E. Cameracci, S. Boochoon, and S. Birchfield. Training deep networks with synthetic data: Bridging the reality gap by domain randomization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 969–977, 2018.

[29] R. Alghonaim and E. Johns. Benchmarking domain randomisation for visual sim-to-real transfer. *arXiv preprint arXiv:2011.07112*, 2020.

[30] J. Tremblay, T. To, B. Sundaralingam, Y. Xiang, D. Fox, and S. Birchfield. Deep object pose estimation for semantic robotic grasping of household objects. *arXiv preprint arXiv:1809.10790*, 2018.

[31] T. E. Lee, J. Tremblay, T. To, J. Cheng, T. Mosier, O. Kroemer, D. Fox, and S. Birchfield. Camera-to-robot pose estimation from a single image. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9426–9432. IEEE, 2020.

[32] I. P. Howard. *Perceiving in depth, volume 1: basic mechanisms*. Oxford University Press, 2012.

[33] D. A. Forsyth and J. Ponce. *Computer vision: a modern approach*. Pearson,, 2012.

[34] W. Luo, A. G. Schwing, and R. Urtasun. Efficient deep learning for stereo matching. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5695–5703, 2016.

[35] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2462–2470, 2017.

[36] N. Mayer, E. Ilg, P. Fischer, C. Hazirbas, D. Cremers, A. Dosovitskiy, and T. Brox. What makes good synthetic training data for learning disparity and optical flow estimation? *International Journal of Computer Vision*, 126(9):942–960, 2018.

[37] J. Zbontar and Y. LeCun. Computing the stereo matching cost with a convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1592–1599, 2015.

[38] F. Yu, V. Koltun, and T. Funkhouser. Dilated residual networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 472–480, 2017.

[39] P. J. Huber. Robust estimation of a location parameter. In *Breakthroughs in statistics*, pages 492–518. Springer, 1992.

[40] A. Kirillov, R. Girshick, K. He, and P. Dollár. Panoptic feature pyramid networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6399–6408, 2019.

[41] T. Hou, A. Ahmadyan, L. Zhang, J. Wei, and M. Grundmann. Mobilepose: Real-time pose estimation for unseen objects with weak shape supervision. *arXiv preprint arXiv:2003.03522*, 2020.

[42] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian. Centernet: Keypoint triplets for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6569–6578, 2019.

[43] M. Pharr, W. Jakob, and G. Humphreys. *Physically based rendering: From theory to implementation*. Morgan Kaufmann, 2016.

[44] T. Yamamoto, K. Terada, A. Ochiai, F. Saito, Y. Asahara, and K. Murase. Development of human support robot as the research platform of a domestic mobile manipulator. *ROBOMECH journal*, 6(1):1–15, 2019.

[45] H. Wang, S. Sridhar, J. Huang, J. Valentin, S. Song, and L. J. Guibas. Normalized object coordinate space for category-level 6d object pose and size estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2642–2651, 2019.

[46] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[47] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013.

[48] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[49] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[50] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar. Hyperband: A novel bandit-based approach to hyperparameter optimization. *The Journal of Machine Learning Research*, 18(1):6765–6816, 2017.

[51] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.